Artificial Intelligence in Medicine (2004) 31, 169–182

ELSEVIER

Artificial Intelligence in Medicine

http://www.intl.elsevierhealth.com/journals/aiim

An evaluation of a system that recommends microarray experiments to perform to discover gene-regulation pathways

Changwon Yoo[a,*], Gregory F. Cooper[b]

[a]420 Social Science, University of Montana, Missoula, MT 59812, USA
[b]Center for Biomedical Informatics, University of Pittsburgh, 8084 Forbes Tower, 200 Lothrop St., Pittsburgh, PA 15213, USA

Received 10 March 2003; received in revised form 14 April 2003; accepted 16 January 2004

KEYWORDS
Causal discovery;
Systems biology;
Causal Bayesian networks;
Microarray study design

Summary The main topic of this paper is modeling the expected value of experimentation (EVE) for discovering causal pathways in gene expression data. By experimentation we mean both interventions (e.g., a gene knockout experiment) and observations (e.g., passively observing the expression level of a ''wild-type'' gene). We introduce a system called GEEVE (causal discovery in Gene Expression data using Expected Value of Experimentation), which implements expected value of experimentation in discovering causal pathways using gene expression data. GEEVE provides the following assistance, which is intended to help biologists in their quest to discover gene-regulation pathways:

- Recommending which experiments to perform (with a focus on ''knockout'' experiments) using an expected value of experimentation method.
- Recommending the number of measurements (observational and experimental) to include in the experimental design, again using an EVE method.
- Providing a Bayesian analysis that combines prior knowledge with the results of recent microarray experimental results to derive posterior probabilities of gene regulation relationships.

In recommending which experiments to perform (and how many times to repeat them) the EVE approach considers the biologist's preferences for which genes to focus the discovery process. Also, since exact EVE calculations are exponential in time, GEEVE incorporates approximation methods. GEEVE is able to combine data from knockout experiments with data from wild-type experiments to suggest additional experiments to perform and then to analyze the results of those microarray experimental results. It models the possibility that unmeasured (latent) variables may be responsible for some of the statistical associations among the expression levels of the genes under study.

To evaluate the GEEVE system, we used a gene expression simulator to generate data from specified models of gene regulation. The results show that the GEEVE system gives better results than two recently published approaches (1) in learning the generating models of gene regulation and (2) in recommending experiments to perform.

*Corresponding author. Tel.: +1-406-243-5605.
E-mail addresses: cwyoo@cs.umt.edu (C. Yoo), gfc@cbmi.upmc.edu (G.F. Cooper).

0933-3657/$ – see front matter © 2004 Published by Elsevier B.V.
doi:10.1016/j.artmed.2004.01.018

1. Introduction

Most research on causal discovery using causal networks has been based on using passive observational data [1—3]. There are limitations in learning causal relationships from observational data only. For example, if the generating process contains a latent factor (confounder) that influences two variables, it can be difficult, if not impossible, to learn the causal relationships between those two variables from observational data alone.

To uncover such causal relationships, a scientist generally needs to design a study that involves manipulating a variable (or variables) and then observing the changes (if any) in other variables of interest. In such an experimental study, one or more variables are manipulated and the effects on other variables are measured. On the other hand, *observational data* result from passive (i.e., non-interventional) measurement of some system, such as a cell. In general, both observational and experimental data may exist on a set of variables of interest. Limited time and funds restrict the number of variables that can be manipulated and the number of *experimental repeats* that can be collected for the control and experimental groups. For example, a molecular biologist who is interested in discovering the causal pathway of the genes involved in galactose metabolism first has to select the genes he or she is interested in understanding at a causal level. These genes are usually selected based on previously published results or by the molecular biologist's personal interest. Many issues are considered in determining the number of experimental repeats to obtain for each variable in the study design. Having more experimental repeats will typically tighten the statistical confidence intervals in the data analysis. Considering available time, budget, and other constraints, the biologist will make a decision about the number of experimental repeats to obtain.

Developing causal analysis methods is a key focus of several fields. In statistics, jointly with medicine, issues related to randomized clinical trials (RCTs) are studied, including methods for finding an optimal number of cases using stopping rules [4—6]. In molecular biology, developing techniques that generate efficient experimental designs for high throughput methods, such as cDNA microarrays, is gaining interest [7,8]. In artificial intelligence, techniques using graphical models have been used to model experimentation and have been applied to suggest the next experiment for causal discovery [9—11].

All these prior approaches have made contributions to efficient causal study design (see Section 2 for details). They are not, however, sensitive to issues of limited resources and experimenter preferences. The research reported here is concerned with developing and evaluating a decision-analytic system that addresses these issues in helping a biologist design and analyze studies of cellular pathways using high throughput sources of data. In particular, this paper concentrates on the design and analysis of cDNA microarray studies for uncovering gene regulation pathways. The fundamental methodology, however, is applicable to analyzing other high throughput data sources, such as the measurement of protein-levels, which is a rapidly developing area of biology.

The GEEVE system uses ideas from different areas of study. GEEVE uses causal Bayesian networks (see Section 2.1) and incorporates an experimenter's preference (see Section 2.2) to give recommendations to the experimenter about designing a gene expression experimental study (see Section 2.3). In this section, we shortly provide background of gene array chips and give an overview of the GEEVE system.

1.1. Gene array chips

Three major gene-expression measurement technologies are currently available for measuring the expression levels of many genes at once. One is called a cDNA microarray, or simply *DNA microarray* [12]; another is called an *oligonucleotide array*, or GeneChip® [13]; and a third technique is called Serial Analysis of Gene Expression (SAGE). We concentrate in this paper on the first two techniques, since they are high throughput methods, whereas SAGE is a more time consuming method. The DNA microarray technique uses user-definable probes[1] of DNA microarray, and the oligonucleotide array uses small oligonucleotide (usually 200 or 300 bases) as factory-built probes.

1.2. Problem description

A gene expression study using DNA microarrays usually involves two major steps. The first step typically consists of performing initial experiments to narrow the set of genes to study in more detail. The experimenter can avoid this first step if he or she already knows the specific set of genes of interest. Since the functions of many genes are not known, the first step is usually necessary.

[1] According to the nomenclature recommended by B. Phimister of *Nature Genetics*, a *probe* is the nucleic acid with known sequence, whereas a *target* is the free nucleic acid sample whose abundance level is being detected.

A number of microarrays will be assigned to hybridize with a pool of controlled cells and experimental cells. By examining the genes that are differentially expressed in these two groups of cells, the experimenter can decide which genes to study further. After choosing those genes, the experimenter has to produce an experimental design for further study how those genes are functionally related to each other.

2. GEEVE system

This chapter describes the issues related to the implementation of the GEEVE system (causal discovery in Gene Expression data using Expected Value of Experimentation). Tong and Koller [11] used a single-case approach to recommend to the experimenter the best possible pairwise relationship for further investigation. In gene expression microarray studies, it may not be practical to perform one experiment at a time. Often it is more efficient to repeat a given experiment multiple times in parallel, rather than to repeat the experiment sequentially over time.

Tong and Koller [11] and Ideker et al. [10] used edge entropy loss functions to search for the next best experiment to perform. This approach can be useful when the experimenter is performing a first-phase study to select the genes without any preference toward the relationships among the genes. After the first-phase study, however, the experimenter will usually have some preference for which genes to study in greater detail. As more gene expression experiments (studies) are performed, the experimenter will refine his or her preferences about the relationships to study in more and more detail. Consequently, a recommendation system that incorporates the preferences of the experimenter seems desirable.

GEEVE allows for repeats of an experiment, and as just mentioned it can be sensitive to an experimenter's preferences for which genes to study. These improvements ostensibly make GEEVE more applicable to real-world design of gene expression experiments. GEEVE also incorporates an efficient causal discovery method that is based on an extension of a causal discovery algorithm [14].

The GEEVE system consists of two modules called the causal Bayesian network update (CBNU) module and the decision tree generation and evaluation (DTGE) module (Fig. 1). The CBNU module uses an algorithm called *Implicit Latent Variable Scoring* (ILVS) method [14] to causally analyze the current microarray data in light of the user's prior beliefs about causal relationships among the genes under study. The DTGE module evaluates a decision tree that was generated based on the results of the CBNU module and the experimenter's preferences, which are expressed with GEEVE as a utility function. Finally (under assumptions) the best possible experiments are recommended to the experimenter. The experimenter then chooses the next experiment to perform, which may or may not be the one suggested by GEEVE. When the results are available, they can be submitted to the CBNU module for a new round of analysis.

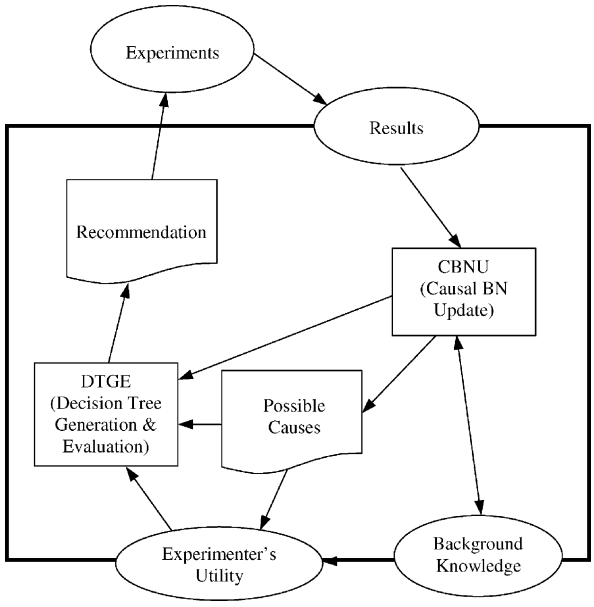

Figure 1 The GEEVE system. The box with the thick line represents the GEEVE system. Boxes in GEEVE represent system modules. Boxes with wavy lines on the bottom represent outputs from GEEVE. The experiments oval is an object that is outside of GEEVE. The ovals on the GEEVE border represent objects that communicate with GEEVE from the outside.

2.1. Updating causal Bayesian networks

This chapter describes a new method to evaluate causal Bayesian networks using a mixture of observational and experimental data. The algorithm described in the current chapter is then incorporated into the GEEVE system.

A causal Bayesian network (or *causal network* for short) is a Bayesian network in which each arc is interpreted as a direct causal influence between a parent node (variable) and a child node, relative to the other nodes in the network [15]. Fig. 2 illustrates the structure of a hypothetical causal Bayesian network structure containing five nodes that represent genes. The probabilities associated with this causal network structure are not shown.

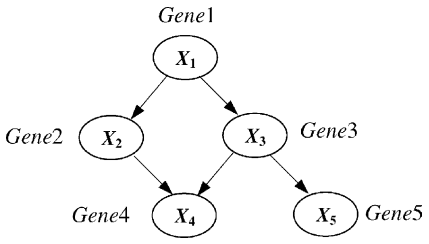

Figure 2 A causal Bayesian network that represents a portion of the gene-regulation pathway for galactose metabolism in yeast.

The causal network structure in Fig. 2 indicates, for example, that the *Gene*1 can regulate (causally influence) the expression level of the *Gene*3, which in turn can regulate the expression level of the *Gene*5. The causal Markov condition gives the conditional independence relationships specified by a causal Bayesian network:

A variable is independent of its non-descendants (i.e., non-effects) given its parents (i.e., its direct causes).

The causal Markov condition permits the joint distribution of the n variables in a causal Bayesian network to be factored as follows [15]:

$$P(x_1, x_2, \ldots, x_n|K) = \prod_{i=1}^{n} P(x_i|\pi_i, K) \quad (1)$$

where x_i denotes a state of variable X_i, π_i denotes a joint state of the parents of X_i, and K denotes background knowledge.

We introduce six equivalence classes (E_1 through E_6) among the structures (Fig. 3). The causal networks in an equivalence class are statistically indistinguishable for any observational and experimental data on X and Y where H represents a latent variable.

Using the previously published structure scoring method [1,16], we introduced the ILVS method to score the six hypotheses in Fig. 3 [14,17]. *Local ILVS Method* (*LIM*) was introduced to score structures with more than pairwise variables [14]. A high level pseudo-code is given in Fig. 4. More detail information of ILVS and LIM could be found at Yoo and Cooper [18] and Yoo [19].

2.2. GEEVE utility model

GEEVE is capable of incorporating an experimenter's utility model [19]. In the research reported in this paper, we did not explore this aspect of GEEVE, because we empirically compare GEEVE's performance to other methods that do not allow modeling utilities flexibly. Instead, we used the following utility assumptions, where E_i^{XY} denotes the node pair X and Y with causal relationship E_i: (1) For all pairs (X, Y), $U(X, Y) = 0.5$, which means that all gene pairs are of equal interest; (2) $U(E_i^{XY}|E_j^{XY}) = 1$ for all $i = j$, which that when the predicted structure E_i^{XY} matches the generating structure E_j^{XY}, the utility is assigned to be the highest possible value (=1.0); (3) $U(E_i^{XY}|E_j^{XY}) = 0.5$ for all E_i^{XY} and E_j^{XY} that have equivalent causal relationships with respect to a latent confounder, that is, E_1^{XY} and E_4^{XY}, E_2^{XY} and E_5^{XY}, and E_3^{XY} and E_6^{XY} are equivalent causal relationships with respect to latent confounder; and otherwise (4) $U(E_i^{XY}|E_j^{XY}) = 0$.

The GEEVE utility for reporting the relationship E_i^{XY} to the user (experimenter) is derived as follows. The weights $w_{ij} = U(E_i^{XY}|E_j^{XY})$ are used as a shorthand notation. The following term is then derived:

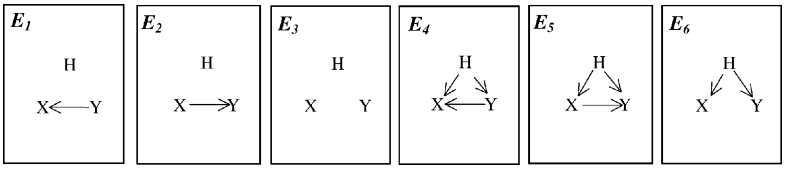

Figure 3 Six local causal hypotheses.

```
For each (X, Y), X ≠ Y and X, Y ∈ {All modeled variables}
  Ω ← Select Correlated Variables of (X, Y) /* |Ω| < k */
  For i = 1 to 6
   S ← Greedy Hill Climbing Structure Search with variables in Ω and X and Y
   /* Perturb S and perform Greedy Hill Climbing Structure Search on S
within the user-defined number of iterations */
   /* Score Ei using ILVS and model averaging */
  EndFor
  /* Normalize score of E1 for (X,Y)*/
EndFor
```

Figure 4 A high level pseudo-code of LIM. Note that S is *a local structure* in which it does not include all modeled variables (recall that the set Ω is limited to include only k variables).

$q_i = \sum_j w_{ij} \cdot P(E_j^{XY}|D,K)$. Finally, the experimenter's utility for discovering a novel and interesting causal relationship is calculated as $q_i \cdot U(X,Y)$.

2.3. Generating a decision tree

Based on the experimenter's utility specification and the causal Bayesian network output (generated by LIM [14] through a local heuristic search and model selection) the GEEVE system builds a decision tree and evaluates it. GEEVE concentrates on pairwise relationships of genes and generates the following decision tree shown in Fig. 5, where R_j represents a pair of genes, np represents the number of pairs among the genes, m represents a maximum measurements that are obtained for an experimental study, ne_h represents the experimental conditions (explained later in this section) to impose for dataset simulation, tE_i represents the situation where the true structure is E_i, and q_i is defined as in Section 2.2.

For the decision tree shown in Fig. 5, assuming that there are at most l states for each variable and assuming there are k variables modeled in LIM's local structure (see Fig. 4), then the number of possible datasets $nd \leq l^{km}$, which is exponential in the number m of microarray experiments (cases). LIM uses a simulation method [20] to make the number of possible datasets manageable. LIM keeps track of the highest scoring local structure given an experimental condition and a dataset D. Using the highest scored local structure, LIM generates possible experimental results such as D'_1, $D'_2, \ldots, D'_{nd}$ [20].

The computation of the decision tree evaluation is exponential in the number of microarray experiments (cases). Therefore, we need an approximation method to evaluate the decision tree. Several different approximation methods are available with some assumptions [21,22].

Heckerman et al. [21] introduced a non-myopic approximation method assuming that for a large decision tree, the central limit theorem holds. The method was non-myopic in the number of chance nodes but not in the number of decision nodes.

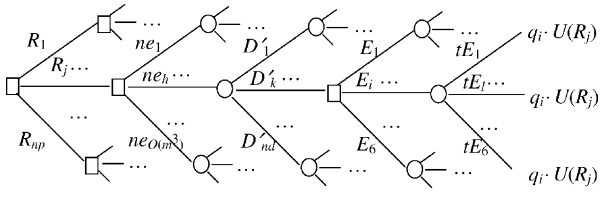

Figure 5 Specifying the experimental condition decision branch.

Chavez and Henrion [22] assumed additive expected utility independence and used linear regression to estimate the expected value of perfect information (EVPI) and expected value of information (EVI). However, Heckerman et al. [21] and Chavez and Henrion [22] approximations are not suitable with large number of decision branches because they assume binary decision nodes. Thus, we use a random heuristic search to approximate the expected value of experimentation (EVE), as explained next.

GEEVE models possible experimental conditions for node pair X and Y as (1) passively observing X and Y, e.g., a wild-type experiment; (2) complete intervention of X (or Y), e.g., a knockout experiment. If a maximum number of experiments repeats that experimenter can request is m, the number of possible experimental conditions to search is $O(m^3)$. Also the number of datasets nd that are possibly generated by an experiment condition, nd, is exponential in m. For an efficient search, GEEVE defines operations on the number of microarray experiments to measure on gene X and Y. Let set $ne_h = (m_O, m_X, m_Y)$, represent the experimental condition where the first element m_O is the number of measurements in which both X and Y are both passively observed (e.g., a ''wild-type'' measurement); the second element m_X is the number of measurements in which X is manipulated and Y is observed; and the third element m_Y is the number of measurements in which Y is manipulated and X is observed. The operators used in GEEVE's heuristic greedy hill climbing search for the best setting of the parameter vector $ne_h = (m_O, m_X, m_Y)$ are as follows:

- MZ(ne_h, i): set the to ith element in ne_h to zero.
- DH(ne_h, i): decrease the ith element in ne_h by half.
- DD(ne_h, i): double the ith element in ne_h.

Using these operators, GEEVE performs the following heuristic search for the value of the parameters (m_O, m_X, m_Y):

- Step 1: The initial parameter values that are tried in the decision tree as follows:

$$\{m - \lceil \tfrac{2}{3}m \rceil, \lceil \tfrac{1}{3}m \rceil, \lceil \tfrac{1}{3}m \rceil\}, \quad \{0, m - \lceil \tfrac{1}{2}m \rceil, \lceil \tfrac{1}{2}m \rceil\},$$
$$\{m - \lceil \tfrac{1}{2}m \rceil, 0, \lceil \tfrac{1}{2}m \rceil\}, \quad \{m - \lceil \tfrac{1}{2}m \rceil, \lceil \tfrac{1}{2}m \rceil, 0\}$$

  $\{m, 0, 0\}$, $\{0, m, 0\}$, and $\{0, 0, m\}$. Choose the experimental condition $ne_h^* = \{m_O^*, m_X^*, m_Y^*\}$ that has highest expected value.
- Step 2: Set ne_h to be ne_h^*.
- Step 3: Apply MZ(ne_h, i), DH(ne_h, i), and DD(ne_h, i) for $i = 1, 2, 3$.

- Step 4: Evaluate expected value for all experimental conditions in Steps 2 and 3 and choose the experimental condition $ne'_h = \{m'_O, m'_X, m'_Y\}$ that has highest expected value. If the expected value of ne'_h is higher than ne^*_h then let ne^*_h be ne'_h and repeat Step 2; otherwise randomly select $ne_h = \{m_O, m_X, m_Y\}$ where $m_O + m_X + m_Y = m$ and go to Step 3 if the repetition is smaller than some user-defined threshold.

The best experiment found by GEEVE when it completes its heuristic evaluation of the decision tree will be the experimental condition $ne_{\max} = \arg\max_{h\in\{1,2,\ldots,O(m^3)\}} ne^*_h$ on the gene pair R_j, where h is the index of ne^*_h that yielded ne^*.

3. Related work

The GEEVE system incorporates an experimenter's preferences into a decision model in order to give recommendations about designing a gene-expression experimental study. The decision model it uses is based on decision theory [23,24]. Many different fields concentrate on study design for causal discovery. Traditionally, in statistics and medicine, research on causal discovery is actively pursued in research on controlled trials [4,5,25]. In computer science, causal discovery is also an active research topic, especially in the machine learning community [1–3,19,26,27]. In biology, recent microarray technologies have fueled a field known as *systems biology*, which seeks to discover causal relationships among a large number of genes and other cellular constituents [28,29]. In this section, we will review work related to this paper, concentrating especially on the fields just mentioned.

3.1. Genetic pathway models

Before describing pathway models, we first place them in the context of gene clustering methods, which have been very popular the last few years. Indeed most of the early work on gene expression data analyses used clustering methods. Gene expression levels that were measured by cDNA microarray in the yeast cell-division cycle were analyzed for the first time using a cluster analysis [28]. A cluster analysis typically searches for groups of genes that show similar expression pattern among different experimental conditions. Other analyses followed using similar cluster analyses applied to microarray data [30–32]. Cluster and classification analyses do not necessarily provide causal information, which is at the heart of gene pathway discovery. On the other hand, knowledge of causal pathways can be used to produce a causal clustering of the genes.

Tsang [33] and Dutilh [34] each give a review of genetic networks. A review that is focused more on modeling methods is given by de Jong [35]. A thorough review based on biological context was published by Smolen et al. [36], who suggested that current microarray techniques are limited in delineating intracellular signaling pathways. Smolen et al. [36] argues that since microarray technology is measuring an average expression level of a gene among millions of cells, there is little we can learn about gene-regulation pathways information from the data. We will discuss this issue in Section 5.2 with respect to latent variable detection.

3.2. Experimentation recommendation models

Computational models of scientific discovery were actively studied in artificial intelligence (in conjunction with psychology) in the late 1980s [37]. In molecular biology in particular, Karp [38] created systems in bacterial gene regulation that could describe the initial conditions of an experiment, generate a hypothesis, and refine it. We will describe additional systems in Sections 3.2.1 and 3.2.2 in more detail because they will be used as points of comparison when evaluating GEEVE in Section 4.

3.2.1. Active learning in Bayesian networks

An extension of supervised learning, *active learning* was applied to learning causal Bayesian networks in scientific discovery [11]. Tong and Koller used edge entropy loss functions and a myopic search in order to recommend the next best experiment to perform. Their main assumptions are: (1) discrete variables only; (2) no missing data; and (2) no modeling of latent (hidden) variables. They modeled manipulation and selection using the manipulation representation in Cooper and Yoo [39].

Tong and Koller applied their algorithm to three Bayesian networks with 5, 8, and 16 nodes respectively. Based on their simulations, they showed that active learning performs better in finding BN structures than randomly choosing of the query nodes.

3.2.2. Entropy score and set covering in Boolean networks

Ideker et al. [10] used binary networks to model the perturbation on a gene network and used entropy loss function to recommend the next best perturbation to perform, where perturbation on a gene means forcing the gene to take a fixed value. They implemented two methods to infer a genetic network built from a gene expression dataset.

To implement the genetic network, they used a deterministic Boolean model. This model is a simplified version of Bayesian networks (see Section 3.1) where all variables are binary and all conditional distribution tables are simply truth tables.

Similar Boolean networks were used to model the experiments involving the gene networks, and the set-covering method was used to recommend the next best experiment for more than one experimental repeat [9]. Karp et al. used a Boolean circuit model of a biological pathway [40] to model experimentation.

4. Evaluation

This section describes an evaluation of the GEEVE system. In the evaluation, we used a simulator to generate gene expression data and compared the performance of the GEEVE system with a system of Tong and Koller [11] (call it TK system) and a system of Ideker et al. [10] (call it ID system), which were described in Section 2. Additionally, we compare GEEVE with GEEVE base line system that restricts the GEEVE system to recommend a case at a time (call it GEEVE_BL system).

4.1. Simulator for the evaluation

Only a few gene expression simulation systems are currently available [41–43]. Limited functions are available in most of the systems because they are in their early development stages. For example, Tomita et al. [41] simulate a cell by developing a computer program shell that can execute any specified cell model. But the system is limited in its (1) available cell models, (2) exporting the gene expression levels to a file, and (3) modeling of measurement errors.

We used the Scheines and Ramsey [42] simulator system (which we will call the SR Simulator) to generate gene expression data. The SR simulator models genes within a cell and incorporates biological variance, such as that due to signal loss or gene mutation, as well as measurement error. The simulator uses a user-defined number of cells in each probe (we set each probe to contain 100,000 cells in this study). It allows measurement at different time points and uses the following so called *Glass function* [44] to update an expression level of a gene X:

$$eX^t = eX^{t-1} + \text{rate}[-eX^{t-1} + F_X(\text{causes\_of}(X^t)\backslash X^{t-1})] + \varepsilon_X \quad (2)$$

where X^t represents the gene X at time t and eX^t represents the gene expression level of the gene X at time t, $0 < \text{rate} \leq 1$, causes_of (X^t) are the direct causes of X^t in the model, ''/'' the set difference operator, ε_X an error term drawn from a given probability distribution, and F_X a binary function specified by the user [44]. Binary functions have been used to model natural phenomena including gene causal pathway [45]. Also note that the model used in this evaluation study contain only a one-stage time-lag, an example of this is shown in Fig. 6, i.e., if a gene has a causal relationship with another gene, it means the relationship is modeled as in Fig. 6.

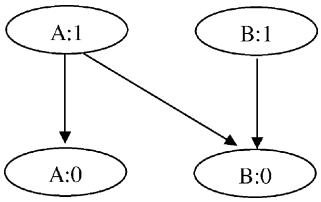

Figure 6 A one-stage time-lag model. A:0 represents the expression level of gene A at current time and A:1 represents the expression level of gene A at one time-step before the current time.

A burn-in period is desirable in applying the SR Simulator. In particular, for the simulated networks discussed in this section (1) it is often after 100 time-lags that the most interesting interactions start among the modeled genes; and (2) the simulated system usually goes into a steady state after 300 time-lags. Therefore we used 100 time-lags for a burn-in period for evaluation study reported here.

4.2. Simulated yeast galactose pathway

In the evaluation we used the SR simulator applied to the yeast galactose metabolic pathways [29] that includes nine galactose genes: *Gal*1, *Gal*2, *Gal*3, *Gal*4, *Gal*5, *Gal*6, *Gal*7, *Gal*10, and *Gal*80. The simulated causal pathway we used is shown in Fig. 7; it only simulates the condition when galactose is provided as a nutrient. The causal pathway was generated based on Ideker et al. [29].

The noise term ε_X in Eq. (2) was estimated from the cDNA microarray dataset provided in Ideker et al. [29] and the rate parameter was estimated as 0.5 by a yeast biologist at our university. F_X in Eq. (2) is defined in Table 1. The function was assessed based on Ideker et al. [29].

4.3. Generated dataset

Initially, we simulated the baseline study, where the experimenter collects an equal number of experimental repeats in different experimental conditions. A dataset of 30 *initial cases* were

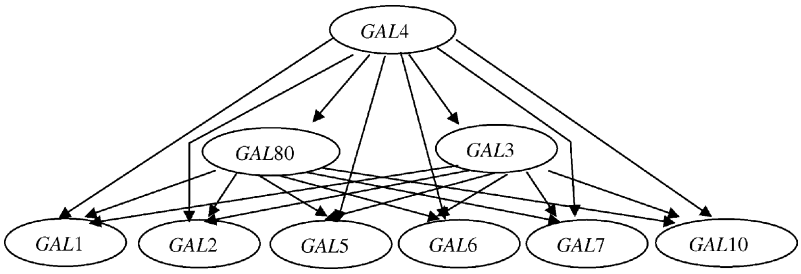

Figure 7 Galactose simulation pathway model.

generated, three experimental repeats (cases) for each of the following 10 experimental conditions: a single wild-type experiment and nine knockout experiments, where a given knockout experiment corresponds to the deletion of one of the nine genes in the simulator model. We generated these initial datasets (and subsequent ones) using a $t = 100$ burn-in period (see Section 4.2). After TK, ID, GEEVE_BL, and GEEVE analyzed the initial dataset, the following steps were iteratively taken with each system:

- Step 1: The system outputs additional knockout experiments to perform.
- Step 2: All of these experiments are performed (using the simulator).
- Step 3: The system analyzes the results of the microarray experiments just performed (combined with the results of any earlier microarray experiments on the same genes under the same experimental conditions).
- Step 4: If the total number of experiments performed thus far is 35 then halt, else go to Step 1.

Let D denote the dataset before Step 1. Then the dataset that is generated after Step 2 is $D \cup$ {results of experiments that a system recommended}. Note that only GEEVE recommends more than one experiment to perform at a time.

The TK, ID, GEEVE_BL, and GEEVE algorithms currently model using discrete variables only, although each could be extended to model with continuous variables as well. The following steps were taken for discretization (i.e., binning) of the simulated gene-expression data (generated from Eq. (2)) that were then used by the algorithms:

(1) Let X_i^* denote the intensity for gene X_i, which serves as an indicator of the expression level of X_i in an experiment in which some gene (not necessarily X_i) was knocked out. Similarly, let rX_i denote the intensity, which is an indicator of the expression level of X_i when no genes were manipulated (wild-type). The relative intensity for gene X_i was calculated as $\log(X_i^*/rX_i^*)$.

Table 1 Definition of F_X that appears in Eq. (2)

	*GAL*4 = 0	*GAL*4 = 1						
*GAL*3	1	0						
	*GAL*4 = 0	*GAL*4 = 1						
*GAL*80	0	1						
	*GAL*4 = 0				*GAL*4 = 1			
	*GAL*3 = 0		*GAL*3 = 1		*GAL*3 = 0		*GAL*3 = 1	
	G80[a] = 0	G80[a] = 1	G80[a] = 0	G80[a] = 1	G80[a] = 0	G80[a] = 1	G80[a] = 0	G80[a] = 1
Other genes GO[b]	0	0	0	0	1	1	0	0

The cause is listed in the columns and the effect in the rows. 0 represents the gene is not expressed and 1 represents the gene is expressed. For example, F_X is defined as (1) if *GAL*4 is expressed then *GAL*3 is supressed and (2) if *GAL*4 is not expressed then *GAL*3 is expressed.
[a] G80 = *Gal*80.
[b] GO = {*Gal*1, *Gal*2, *Gal*5, *Gal*6, *Gal*7, *Gal*10}.

(2) Discretization was performed based on each gene's relative intensity of mean m and standard deviation δ over all relative intensities. All genes were assigned three states: 0 was assigned to any value less than $m-\delta$, 1 was assigned to any value greater than or equal to $m-\delta$ and less than $m+\delta$, and 2 was assigned to any value greater than or equal to $m+\delta$.

For the discretization for the ID system, in Step 2, all genes were assigned two states: 0 was assigned to any value less than m, and 1 was assigned to any value greater than or equal to m.

4.4. Evaluation matrices

IK and ID do not incorporate experimenter's preferences. To make the comparison of IK and ID with GEEVE fair, IK and ID should have a preference module similar to GEEVE. Thus, in this comparison study, we use uniform preference on all gene pairs and causal hypotheses, as described in Section 2.2. The measurement of the performance of TK, ID, GEEVE_BL and GEEVE was based on the two criteria discussed next.

4.4.1. Prediction of the generating causal structure

Let AUROC denote the area under the ROC curve for a prediction that involves genes X and Y. For each gene pair (X, Y), where X, $Y \in \{Gal1, Gal2, Gal3, Gal4, Gal5, Gal6, Gal7, Gal10, Gal80\}$, we calculated two AUROCs. One is for the prediction that X and Y are independent given that they are independent in the simulator's generating model (true positive rate) and given that they are not independent (e.g., their is a causal relationship between them) in the generating model (false positive rate). Another AUROC is for the prediction of causal relationship R between X and Y, given that the causal relationship between X and Y is R (true positive rate) and given that the causal relationship between X and Y is not R (false positive rate). For each algorithm and for each of the two types of AUROC. AUROC is calculated using each dataset in Step 3 of the previous section.

4.4.2. Predictive performance as a function of the number of experiments performed

Using the cost function that was estimated from a molecular biologist collaborator (detail function descriptions are in Section 4.5), we calculated the total cost of the experiments performed by each system. For example, if it takes 2 h to process a microarray chip and it costs US$ 50.00 h^{-1} for such analysis, the total cost (excluding the material costs) is US$ 100.00. As mentioned above, predictive performance is measured using AUROC, which is derived by using the generating relationships as the ground truth. Finally, using the assessed cost function in Section 4.5, we recorded the cost that is associated with attaining a given AUROC and use these factors to derive a unit "*performance accuracy over cost*," which is calculated by dividing the AUROC by the experiment costs in dollars. This unit represents an increased fraction of an AUROC per dollar cost. We plot the AUROC over cost as a function of the number of experiments performed.

In analyzing a given set of data with a given system, we ran the system for up to 2 h for the following reason. A running time of less than 2 h showed relatively high variance on AUROC to the variance on AUROC of running time over 2 h. Furthermore, a running time of 3—4 h showed similar variance on AUROC to that of 2 h running time. We used a 500 MHz dual processor Linux machine to set up the appropriate parameters for each system to run approximately 2 h. For the entire experiment, we used the Linux machine, a 400 MHz Microsoft Windows 2000 machine, and a 266 MHz Microsoft Windows NT machine. All programs were compiled with gnu C++ on the Linux machine and with Microsoft Visual C++ on the Windows machines. The total running time was approximately as follows: 35 additional experiments $\times$ 2 h per system $\times$ 4 systems per experiment $=$ 280 h.

4.5. Results

Results of the predictive performance of each system is shown in Fig. 8(a) and (b). The X-axis represents the number of experiments performed (using simulation) and analyzed by the system. Recall that except for GEEVE, all other systems recommend one microarray experiment at a time. This is why the GEEVE plots are disconnected in Fig. 8. For example, in Fig. 8, GEEVE requests two microarray experiments after it analyzes 15 microarray experiments; after analyzing 17 microarray experiments, it again requests two microarray experiments. Error bars of the AUROC in the figure were calculated using the bootstrap method described in Efron [46]. In particular, for an AUROC curve for a given system, that systems 36 predictions were randomly selected with replacement and this procedure was performed 2000 times. The error bars each represent a 95% confidence interval around a given AUROC.

As shown in Fig. 8(a), there is no system that dominates in predicting the correct independence

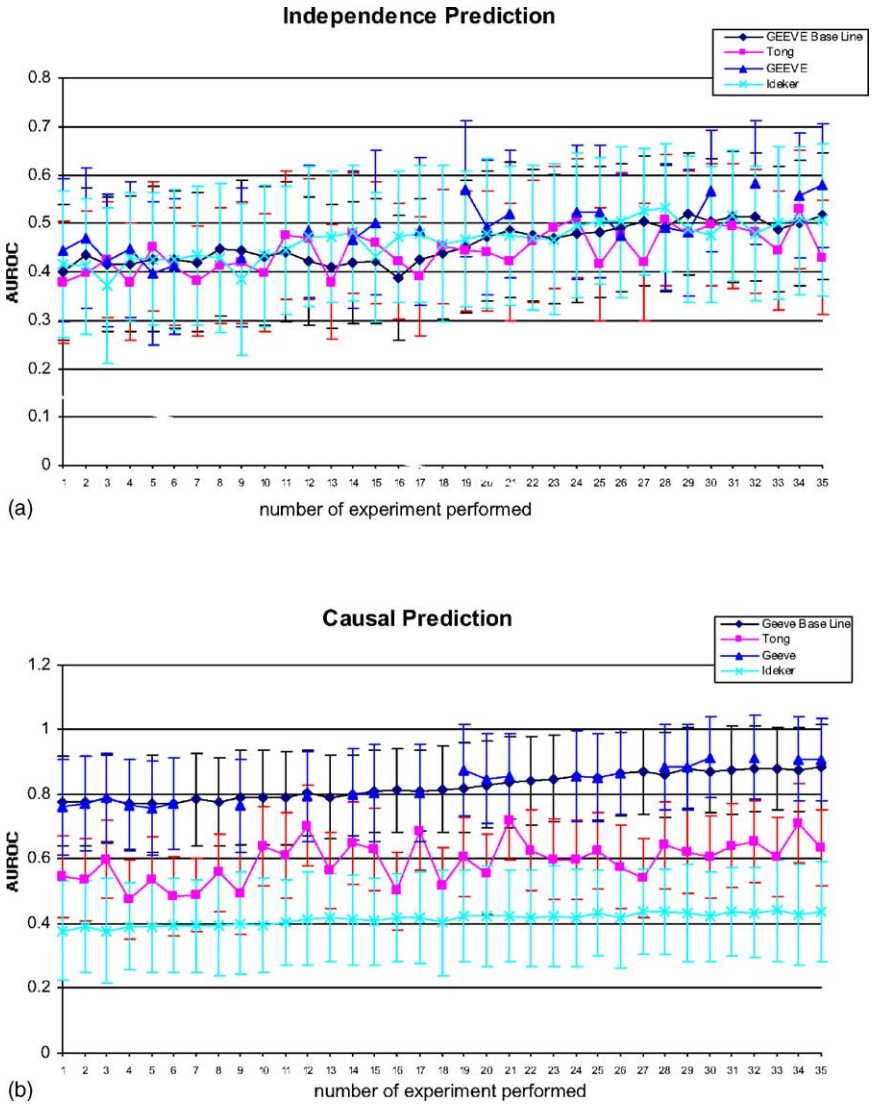

Figure 8 Area under ROC curve (AUROC) as a function of the number of experimental cases performed (via simulation) and used to assess relationships among the variables. Each bar represents a 95% confidence interval. (a) AUROC of independence relationships prediction. (b) AUROC of causal relationships prediction.

relationships. Note that we have to consider the cost to process a cDNA microarray chip. A large portion of the cost is the technician's time to process the microarray chip. Processing one microarray chip at a time is much more costly than batch-processing several microarray chips at a time. This is because there are many steps to take to analyze a cDNA microarray chip, and each step takes a long time to complete.

Consider the following scenario for a given experiment that involves a microarray: (1) perform experiment ξ, (2) analyze the microarray chip results of experiment ξ; (3) based on the results, determine the next experiment ξ' to perform; (4) perform the experiment ξ' with a microarray chip; and (5) analyze the microarray chip results of experiment ξ'. In this scenario, a technician analyzed two microarray chips in all. Now consider the alternate scenario where the technician performs experiments ξ and ξ' together and analyzes the resulting two microarray chips in a batch mode.

Typically, analyzing two chips together will take less of the technician's time than analyzing two chips in series, particularly if $\xi = \xi'$, that is, ξ' is simply a repeat of ξ. The downside, however, is that in doing two experiments at the same time, we are not able to use the results of the first experiment to help tailor which experiment to perform second.

There are different suggested protocols to analyze a microarray chip [47]. It usually takes 16 h (2 days) of a technician's time to produce and analyze one microarray chip. It will usually take 20 and 24 h for him or her to analyze two and three cDNA microarray chips at once, respectively (4 h for

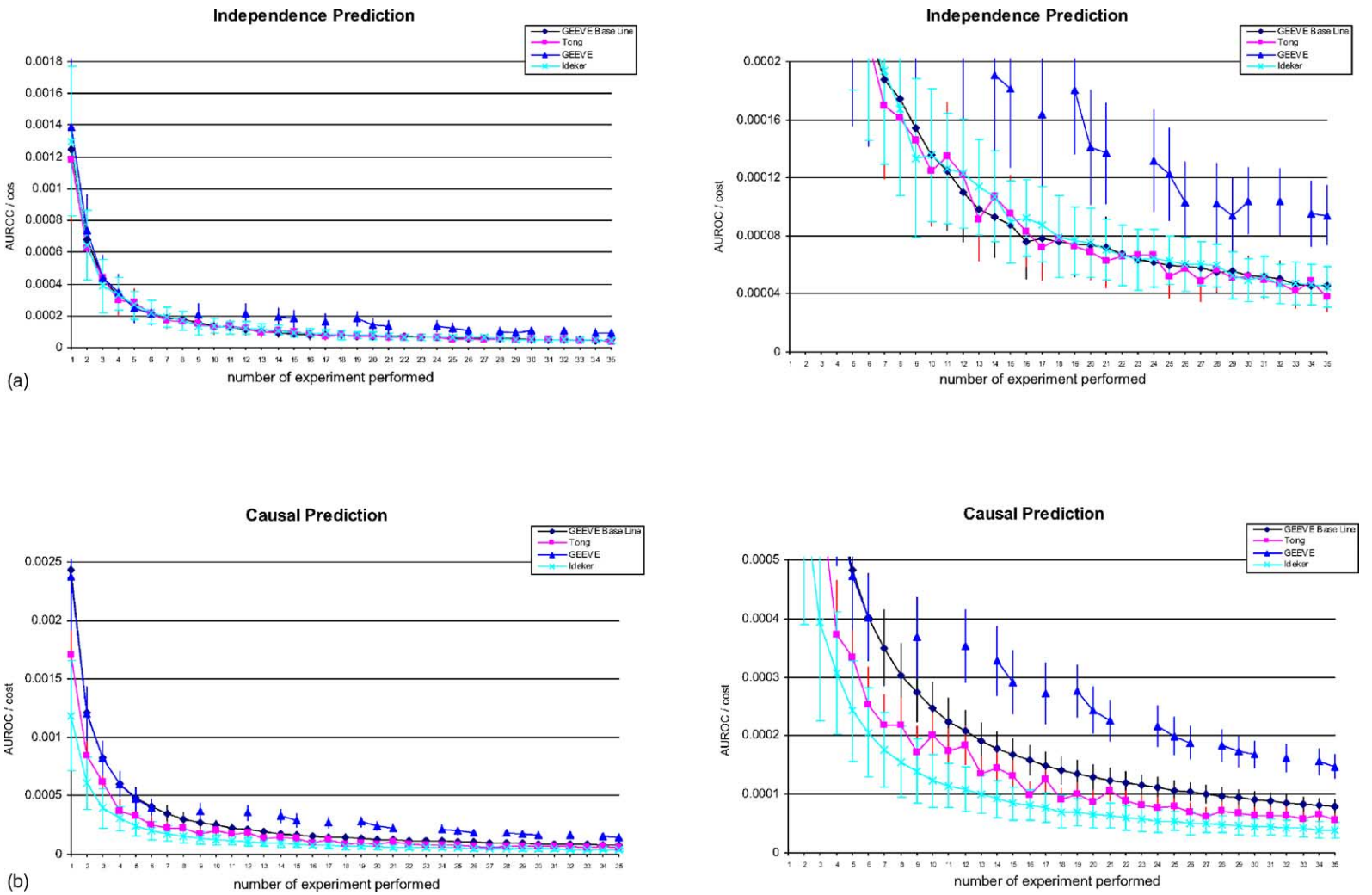

Figure 9 AUROC per cost calculation. Each bar represents a 95% confidence interval. The *X*-axis represents the number of microarray experiments that were suggested by an algorithm and then performed by way of simulation. (a) AUROC/cost of independence relationships prediction. The left graph shows the overall plot and the right graph shows the left graph's lower right plot in detail. (b) AUROC/cost of causal relationships prediction. The left graph shows the overall plot and the right graph shows the left graph's lower right plot in detail.

each additional microarray chip). This is because it usually takes 4 h to finish the first step, extracting DNA. If the technician earns US$ 20 h^{-1}, the costs involved in analyzing two chips in the two different scenarios are: (1) US$ 640 to analyze one chip at a time [(16 h $\times$ 2) $\times$ US$ 20]; and (2) US$ 400 to analyze two chips at once (20 h $\times$ US$ 20). Similarly, the costs involved in analyzing three chips are: (1) US$ 960 to analyze one chip at a time [(16 h $\times$ 3) $\times$ US$ 20]; and (2) US$ 480 to analyze three chips at once (24 h $\times$ US$ 20). We used these cost assumptions in the analyses that follow.

Under these cost assumptions, we can calculate AUROC per dollar, which is shown in Fig. 9. It is clear that under these cost assumptions, GEEVE outperforms the other systems in both causal and independence predictions.

In summary, Fig. 8(b) shows that GEEVE consistently perform better than the ID and TK systems in correctly predicting causal relationships. Fig. 9(b) shows that GEEVE and GEEVE_BL outperform the ID and TK systems in predictive performance per dollar. GEEVE outperforms GEEVE_BL in Figs. 8(b) and 9(b) because GEEVE recommends more than one microarray experiment (case) at a time.

5. Conclusions

Systems biology emphasizes large scale discovery of the *interactions* of genes, proteins, and other cell elements. Systems biology is confronted with a huge number of interactions, not the least of which is the interaction of genes. There are challenges in designing high throughput experiments, such as cDNA microarrays, and for analyzing the high volume of data generated by those experiments in order to discover gene regulation networks. Intrinsically, these issues are causal in nature. We have introduced a new causal analysis method along with a computer system that uses that method to recommend the gene-regulation experiments to perform.

Unlike clinical radomized controlled trials, where an experimenter is interested in the causal relationship of a handful variables (e.g., an experimenter is interested in a new drug and its treatment effect) in systems biology an experimenter is usually interested in the causal relationships among thousands of entities, such as genes. Different approaches are needed in systems biology for causal discovery and experimental design recommendation. This paper has explored one such approach. In the remainder of this section, we summarize the contributions made by this paper and then discuss open problems.

5.1. Local causal search with experimentation recommendations

We developed a system called GEEVE (causal discovery in Gene Expression data using Expected Value of Experimentation) that incorporates an experimenter's preferences regarding which genes to study in order to discover causal relationships among those genes. Among the genes of interest, GEEVE models their likely causal relationships, based on prior biological knowledge and experimental data.

Experiments provide benefit in terms of information, but they also have costs in terms of human labor and the laboratory costs. Considering preferences, costs, and a current model of causal relationships, GEEVE recommends the most cost-effective experiment it can find in its search of the space of experiments.

For evaluation, we modeled and simulated a portion of the yeast galactose metabolic pathway. Using the yeast galactose pathway simulator to generate simulated microarray data, we showed that GEEVE predictions (area under ROC curve) were better (although not highly statistically significantly so) than two other state-of-the-art methods recently described in the literature. When we applied the cost function that was assessed from the biologists and calculated the area under ROC curve as a function of experimental cost, GEEVE showed performance that was statistically significantly different than the other two recommendation systems.

5.2. Future work and open issues

Regarding external experimental conditions, such as nutrient conditions, they could be modeled as exogenous variables in a causal Bayesian network. Currently, GEEVE models only experiments that involve wild-type gene levels and single gene knockouts. In the future, more general experiments, such as over-expression experiments, more than one gene knockout and so forth, should be modeled.

Regarding modeling the time course of gene expression, and determining precisely when to sample cells during experimentation, temporal Bayesian networks appear a natural choice [48,49]. It will be interesting to explore models that use both continuous and discrete variables within temporal Bayesian networks. Temporal Bayesian networks also provide one approach to modeling gene regulation feedback. The six pairwise causal hypotheses used in this research could be extended to model such feedback. This is an important issue for future

research because feedback is widely observed in many cellular pathways.

Currently, GEEVE only generates decision trees based on the discovery of pairwise gene relationships. More generally, R_j in Fig. 5 (Section 2.3) should include more than pairwise relationships. Doing so will allow GEEVE to (1) model beyond a single gene perturbation experiments, such as a knockout of two or more genes at a time; and (2) incorporate (in the decision tree) the effects on other genes besides genes (X, Y) when gene X (or Y) is perturbed.

We have also introduced a causal discovery system that can score latent structures. Since the most closely related prior methods assume no latent variables, there is no straightforward way to evaluate GEEVE's prediction of latent structures with these other methods. Also since cDNA microarray is measuring the average expression level of millions of cells, the variance that we observe in the levels (when an experiment is repeated several times) is due almost entirely to measurement error and not to biological variation [50]. Biological variation is needed to discover latent structure, certainly with LIM, and we believe with any method. Measuring the expression level of genes under various experimental conditions (e.g., measuring at different time points or in different temperatures) can provide biological variation among groups of cells; it is an open question how helpful biological variation of this particular variety will be in discovery of latent structure.

Another way to obtain biological variation in gene expression would be to measure gene expression at the level of a single cell. Such measurements will require new technology. We anticipate that such methods will be developed within the next decade. If so, the methods in this paper will be applicable to suggesting when latent factors (such as unknown proteins) may be influencing two or more specific genes.

Ideker et al. [29] describe four steps in discovering causal pathways among the genes: (1) gather and formulate the current knowledge about the genes and their pathways; (2) design and perform experiments; (3) analyze the data from the experiments; and (4) formulate new hypotheses to explain the analysis results not predicted by Step 1 and then repeat Steps 2, 3, and 4. There are many open issues in how to complete this loop. The soundness of microarray measurements needs to be studied further, e.g., studying the relationship between mRNA levels and protein expression levels, and studying and quantifying the various sources of measurement error related to detecting gene expression levels. Other open issues include detecting genes and their promoter regions from sequence information, compiling known gene regulatory knowledge (and other cell-network knowledge) from the literature, and standardizing causal pathway representations.

Acknowledgements

This research has been partially supported by grants the National Science Foundation (IIS-9812021), the National Cancer Institute, the Mellon Fellowship of University of Pittsburgh, the National Institute of Health, and the National Aeronautics and Space Administration (NRA2-37143).

References

[1] Cooper GF, Herskovits E. A Bayesian method for the induction of probabilistic networks from data. Machine Learning 1992;9:309—47.
[2] Heckerman D. A Bayesian approach to learning causal networks. In: Proceedings of the Conference on uncertainty in artificial intelligence. Morgan Kaufmann; 1995.
[3] Spirtes P, Glymour C, Scheines R. Causation, prediction, and search. 2nd ed. Cambridge, MA: MIT Press; 2000.
[4] Spiegelhalter DJ, Freedman LS, Parmar MKB. Bayesian approach to randomized trials. J R Stat Soc 1994;157(Part 3):357—416.
[5] Berry DA, Stangl DK. Bayesian methods in health-related research. In: Berry DA, Stangl DK, editors. Bayesian Biostatistics. New York: Marcel Dekker; 1996. p. 3—66.
[6] Friedman LM, Furberg CD, DeMets DL. Sample size. In: Fundamentals of clinical trials. 3rd ed. St. Louis: Mosby-Year book; 1996. Chapter 7. p. 94—129.
[7] Karp PD et al. Integrated pathway/genome database and their role in drug discovery. Trends Biotechnol 1999;17(7): 275—81.
[8] Kerr MK, Churchill GA. Experimental design for gene expression microarrays. Biostatistics 2001;2:183—201.
[9] Karp RM, Stoughton R, Yeung KY. Algorithms for choosing differential gene expression experiments. Res Comput Biol 1999.
[10] Ideker T, Thorsson V, Karp RM. Discovery of regulatory interactions through perturbation: inference and experimental design. In: Pacific Symposium Biocompution. 2000.
[11] Tong S, Koller D. Active learning for structure in Bayesian networks. In: International Joint Conference on Artificial Intelligence. Seattle, WA; 2001.
[12] Brown PO, Botstein D. Exploring the new world of the genome with DNA microarrays. Nat Genet 1999;21(Supplement):33—7.
[13] Lipshutz RJ et al. High density synthetic oligonucleotide arrays. Nat Genet 1999;21(Supplement):20—4.
[14] Yoo C, Cooper G. Discovery of gene-regulation pathways using local causal search. In: AMIA. San Antonio, Texas, 2002.
[15] Pearl J. Probabilistic reasoning in intelligent systems. In: Brachman RJ, editor. Representation and reasoning. San Mateo, CA: Morgan Kaufmann; 1988.
[16] Heckerman D, Geiger D, Chickering D. Learning Bayesian networks: the combination of knowledge and statistical data. Machine Learning 1995;20:197—243.

[17] Yoo C, Thorsson V, Cooper GF. Discovery of a gene-regulation pathway from a mixture of experimental and observational DNA microarray data. In: Pacific Symposium on biocomputing. Maui, Hawaii: World Scientific; 2002.
[18] Yoo C, Cooper G. Causal discovery of latent-variable models from a mixture of experimental and observational data. In: CBMI Research Report CBMI-173. Pittsburgh, PA: Center for Biomedical Informatics; 2001.
[19] Yoo C. Expected value of experimentation in causal discovery from gene expression studies. Ph.D. dissertation, 2002.
[20] Henrion M. Propagating uncertainty in Bayesian networks by probabilistic logic sampling. In: Lemmer JF, Kanal LN, editors. Uncertainty in artificial intelligence 2. North-Holland: Amsterdam; 1988. p. 149—63.
[21] Heckerman D, Horvitz E, Middleton B. An approximate nonmyopic computation for value of information. In: Proceedings of the Seventh Conference on uncertainty in artificial intelligence. 1991.
[22] Chavez T, Henrion M. Efficient estimation of the value of information in Monte Carlo models. In: Uncertainty in artificial intelligence. 1994.
[23] von Neumann J, Morgenstern O. Theory of games and economic behavior. Princeton NJ: Princeton University Press; 1944.
[24] Keeney RL, Raiffa H. Decisions with multiple objectives: preference and value tradeoffs. New York: John Wiley; 1976.
[25] Achcar JA. Use of Bayesian analysis to design of clinical trials with one treatment. Commun Stat Theory Methods 1984;13:1693—707.
[26] Pearl J. Causality: models, reasoning, and inference. Cambridge, UK: Cambridge University Press; 2000.
[27] Heckerman D, Meek C, Cooper GF. A Bayesian approach to causal discovery. In: Glymour C, Cooper GF, editors. Computation, causation, and discovery. Menlo Park, CA: AAAI Press; 1999. p. 141—65.
[28] Spellman PT et al. Comprehensive identification of cell cycle-regulated genes of the yeast *Saccharomyces cerevisiae* by microarray hybridization. Mol Biol Cell 1998;9: 3273—97.
[29] Ideker T et al. Integrated genomic and proteomic analysis of a systematically perturbed metabolic network. Science 2001;292:929—34.
[30] Michaels GS, et al. Cluster analysis and data visualization of large-scale gene expression data. Pacific Symposium on biocomputing. 1998.
[31] Herwig R et al. Large-scale clustering of cDNA-fingerprinting data. Genome Res 1999;9:1093—105.
[32] Golub TR et al. Molecular classification of cancer: class discovery and class prediction by gene expression monitoring. Science 1999;286:531—7.
[33] Tsang J. Gene expression, DNA arrays, and genetic network. In: Unpublished manuscript, Bioinformatics Laboratory at University of Waterloo; 1999.
[34] Dutilh B. Gene networks from microarray data. In: Unpublished manuscript. Literature thesis at Utrecht University; 1999.
[35] de Jong H. Modeling and simulation of genetic regulatory systems: a literature review. J Comput Biol 2002;9(1): 67—103.
[36] Smolen P, Baxter DA, Byrne JH. Modeling transciptional control in gene networks—methods. Bull Math Biol 2000; 62:247—92.
[37] Shrager J, Langley P. In: Shrager J, Langley P, editors. Computational models of discovery and theory formation. San Mateo, CA: Morgan Kaufman; 1990.
[38] Karp PD. Hypothesis formation as design. In: Shrager J, Langley P, editors. Computational models of discovery and theory formation. San Mateo, CA: Morgan Kaufman; 1990. p. 276—317.
[39] Cooper GF, Yoo C. Causal discovery from a mixture of experimental and observational data. In: Proceedings of the Conference on uncertainty in artificial intelligence. Morgan Kaufmann; 1999.
[40] Akutsu T, Miyano S, Kuhara S. Identification of genetic networks from a small number of gene expression patterns under the Boolean network model. In: Pacific Symposium on biocomputing. Hawaii; 1999.
[41] Tomita M et al. E-CELL: software environment for whole cell simulation. Bioinformatics 1999;15(1):72—84.
[42] Scheines R, Ramsey J. Gene simulator. In: Available at: http://www.phil.cmu.edu/tetrad/.2001.
[43] Saavedra R, Glymour C. A regulatory network simulator. In: Simulator based on (Yuh et al., 1998) under development. 2001.
[44] Edwards R, Glass L. Combinatorial explosion in model gene networks. Chaos 2000;10:691—704.
[45] Kauffman S. Origins of order—self-organization and selection in evolution. Oxford University Press; 1993.
[46] Efron B. The jacknife, the boostrap and other resampling plans. Soc Ind Appl Math 1982;1092:2—5.
[47] Hegde P et al. A concise guide to cDNA microarray analysis. Biotechniques 2000;29(3):548—62.
[48] Murphy K, Mian S. Modelling gene expression data using dynamic Bayesian networks. In: Technical report, U.B. Department of Computer Science; 1999.
[49] Friedman N, et al. Using Bayesian networks to analyze expression data. J Computat Biol 2000.
[50] Spirtes P, Glymour C, Scheines R. Constructing Bayesian network models of gene expression networks from microarray data. In: To appear in the Proceedings of the Atlantic Symposium on computational biology. Genome Information Systems and Technology; 2001.